Searching for Stationarity

Philip Anderson; panders2@tamu.edu

```
library("astsa")
library("DescTools")
```

```
## Warning: package 'DescTools' was built under R version 3.4.4
```

```
library("GGally")
```

```
## Warning: package 'GGally' was built under R version 3.4.4
```

```
library("tseries")
```

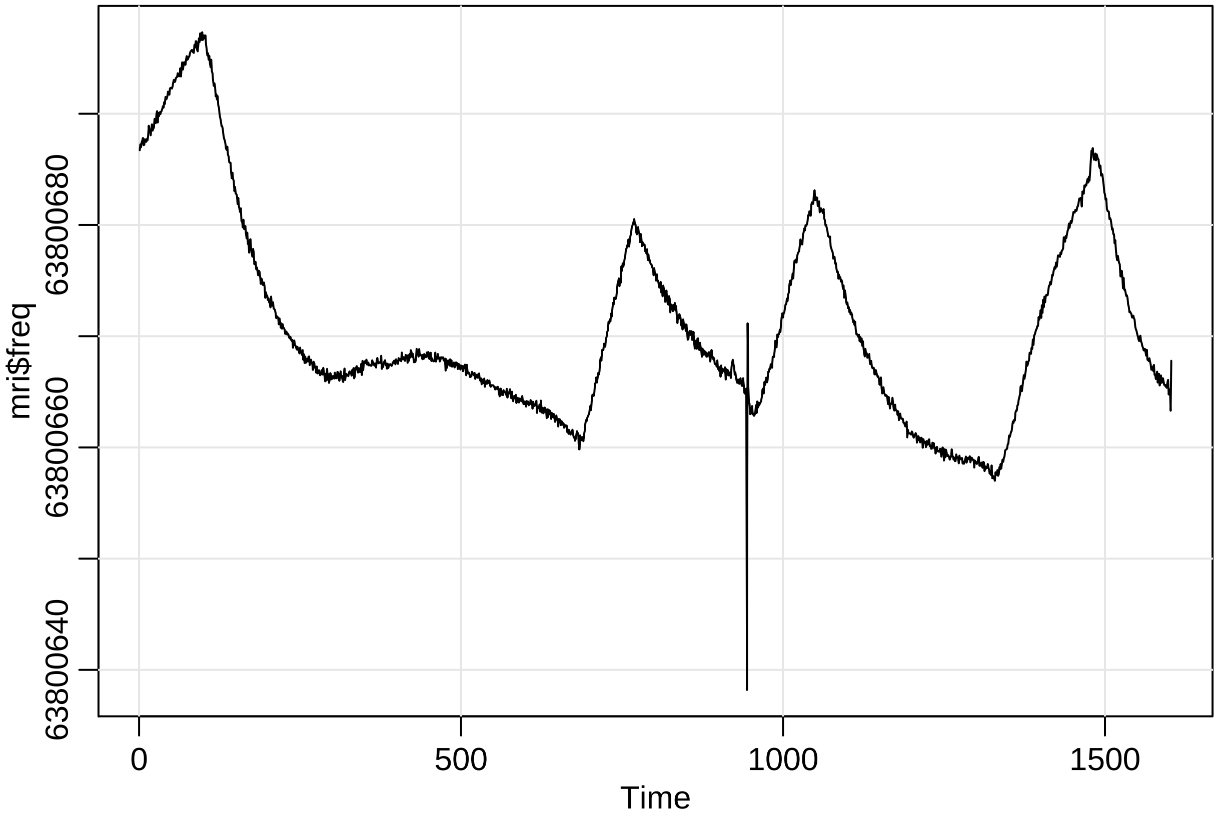
```
## Warning: package 'tseries' was built under R version 3.4.4
```

Objective: I'm going to mess around with some of the Week 3 material - basically looking to see if I can get the autocorrelation function to have no significant lags.

```
# read in the data
mri <- read.csv("/Users/panders2/Documents/schools/tamu/stat_626/project/stat_626_proj/mri_dat_one.csv"
# update field names
names(mri) <- c("hour", "minute", "freq", "int_pressure", "atm_pressure", "tot_pressure", "tesla")
```

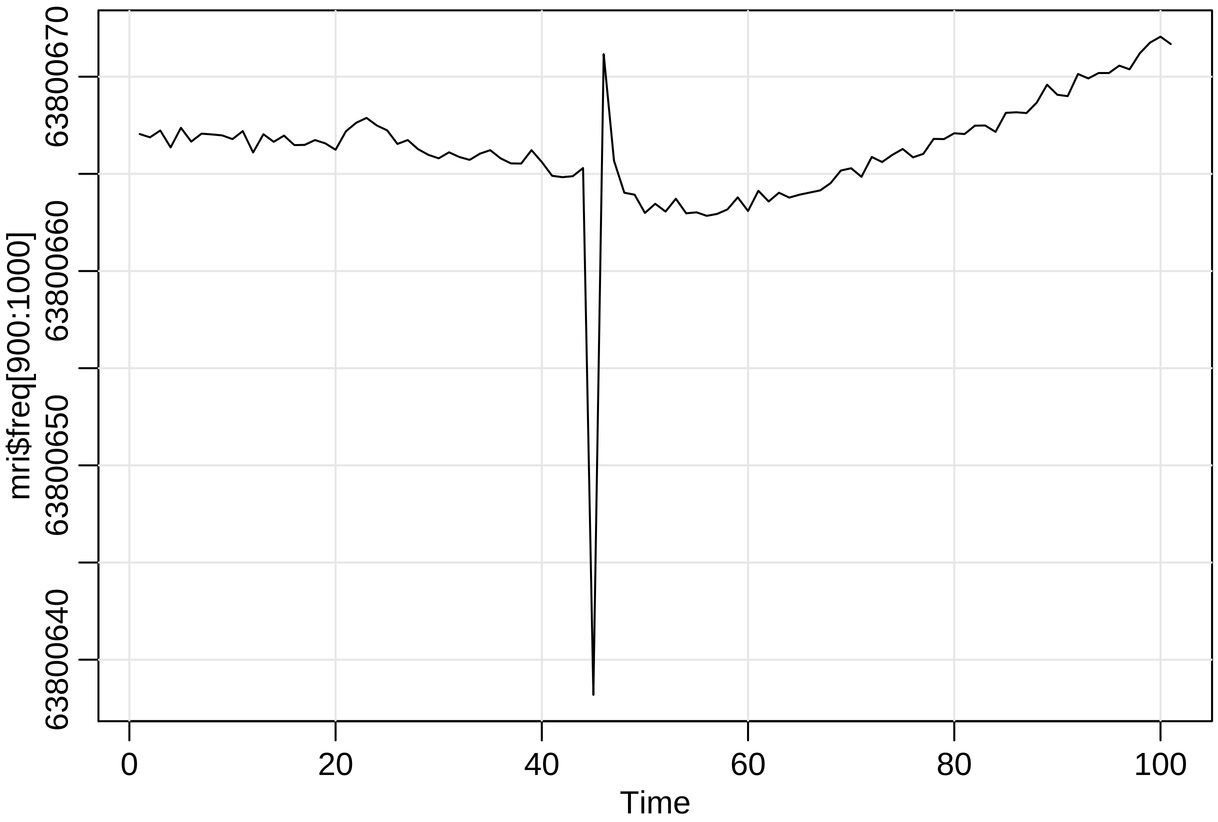
With an initial look at the data, I think we need to smooth it to eliminate the abberration at time index ~950.

```
astsa::tsplot(mri$freq)
```



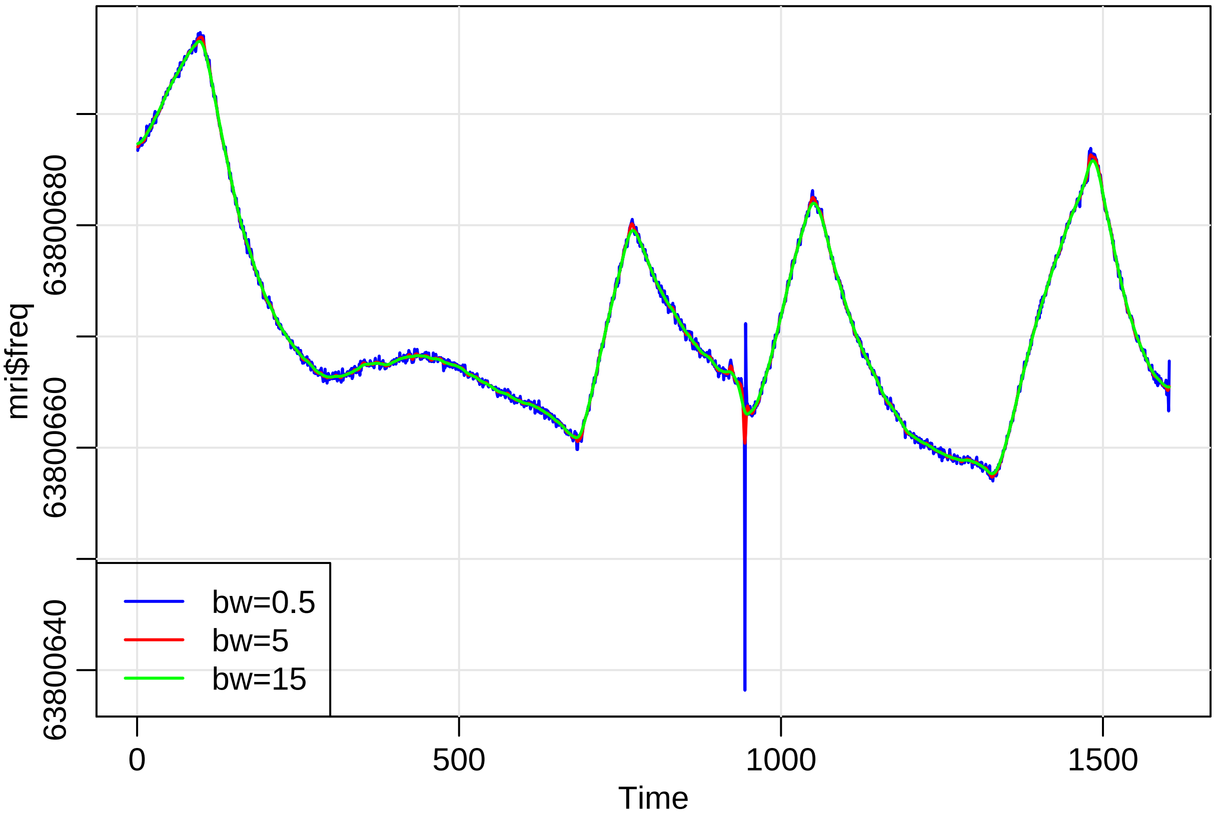
Zoom in:

```r
astsa::tsplot(mri$freq[900:1000])
```



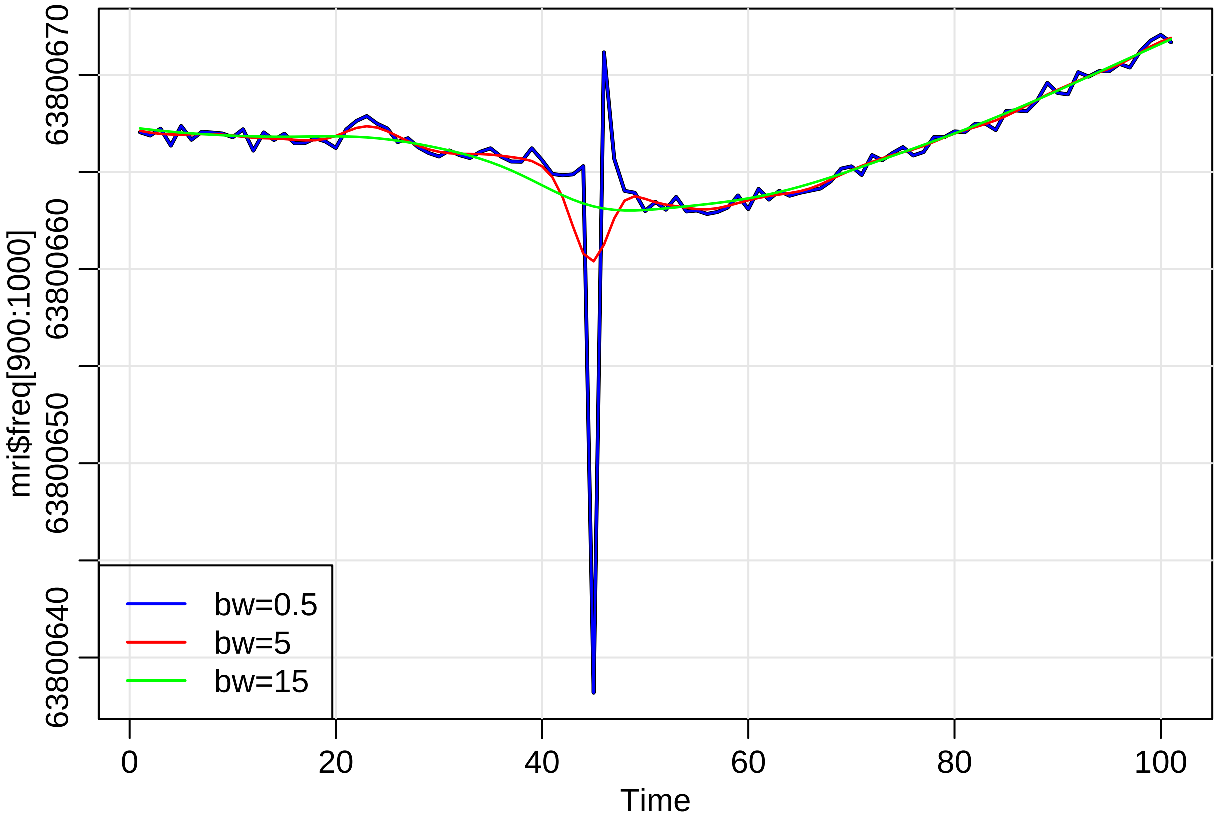
I don't know how much of an impact this outlier will have on any sort of models we build, but it probably isn't good. Regardless, taking care of it will be a good exercise in smoothing, so let's do it anyway. The *ksmooth* function used by the book does not appear to have automatic bandwidth selection, so let's try several and see what happens.

```r
astsa::tsplot(mri$freq)
lines(ksmooth(time(mri$freq), mri$freq, kernel="normal", bandwidth=0.5), lwd=1.5, col="blue")
lines(ksmooth(time(mri$freq), mri$freq, kernel="normal", bandwidth=5), lwd=1.5, col="red")
lines(ksmooth(time(mri$freq), mri$freq, kernel="normal", bandwidth=15), lwd=1.5, col="green")

legend("bottomleft", c("bw=0.5", "bw=5", "bw=15"), lwd=rep(1.5, 3) , col=c("blue", "red", "green"))
```



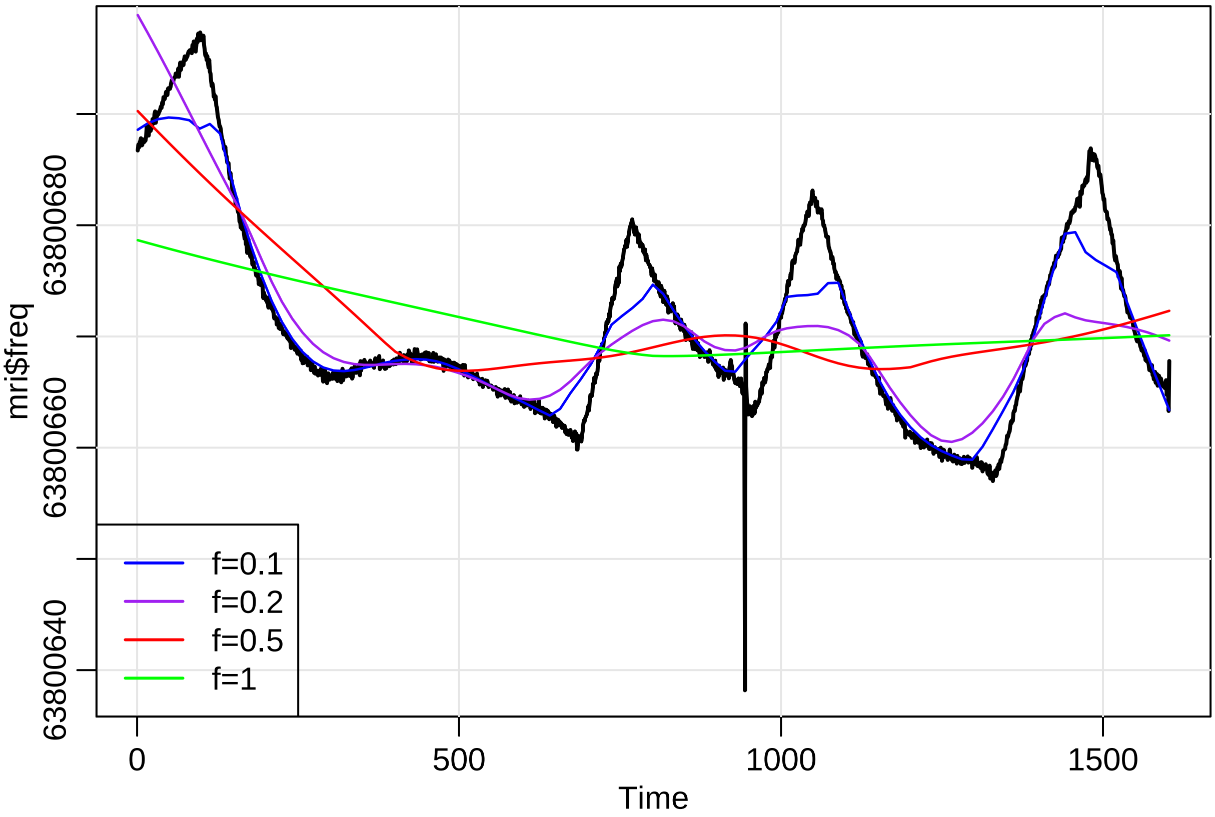
Focusing on just the problem area, it seems that a higher bandwidth does the best job at smoothing this area out. There may be some concern about over-smoothing, but we can proceed as is. Once again, let's zoom in.

```
# (I believe) kernel smoothing utilizes the entire array for producing estimates, so we will
# need to make new variables on the entire set before zooming

mri$freq_point5 <- ksmooth(time(mri$freq), mri$freq, kernel="normal", bandwidth=0.5)$y
mri$freq_5 <- ksmooth(time(mri$freq), mri$freq, kernel="normal", bandwidth=5)$y
mri$freq_15 <- ksmooth(time(mri$freq), mri$freq, kernel="normal", bandwidth=15)$y

astsa::tsplot(mri$freq[900:1000], lwd=2)
lines(mri$freq_point5[900:1000], lwd=1.25, col="blue")
lines(mri$freq_5[900:1000], lwd=1.25, col="red")
lines(mri$freq_15[900:1000], lwd=1.25, col="green")

legend("bottomleft", c("bw=0.5", "bw=5", "bw=15"), lwd=rep(1.5, 3) , col=c("blue", "red", "green"))
```



Kernel smoothing is one viable option - localized regression is another. Let's do that and compare. Like the kernel bandwidth parameter, the lowess regressor incorporates a parameter that influences smoothness, (f). It simply measures what proportion of the array points we should use to influence the smooth at each estimation point.

```
astsa::tsplot(mri$freq, lwd=2)
lines(lowess(mri$freq, f=0.1), lwd=1.25, col="blue")
lines(lowess(mri$freq, f=0.2), lwd=1.25, col="purple")
lines(lowess(mri$freq, f=0.5), lwd=1.25, col="red")
lines(lowess(mri$freq, f=1), lwd=1.25, col="green")

legend("bottomleft", c("f=0.1", "f=0.2", "f=0.5", "f=1"), lwd=rep(1.5, 5) , col=c("blue", "purple",  "re
```

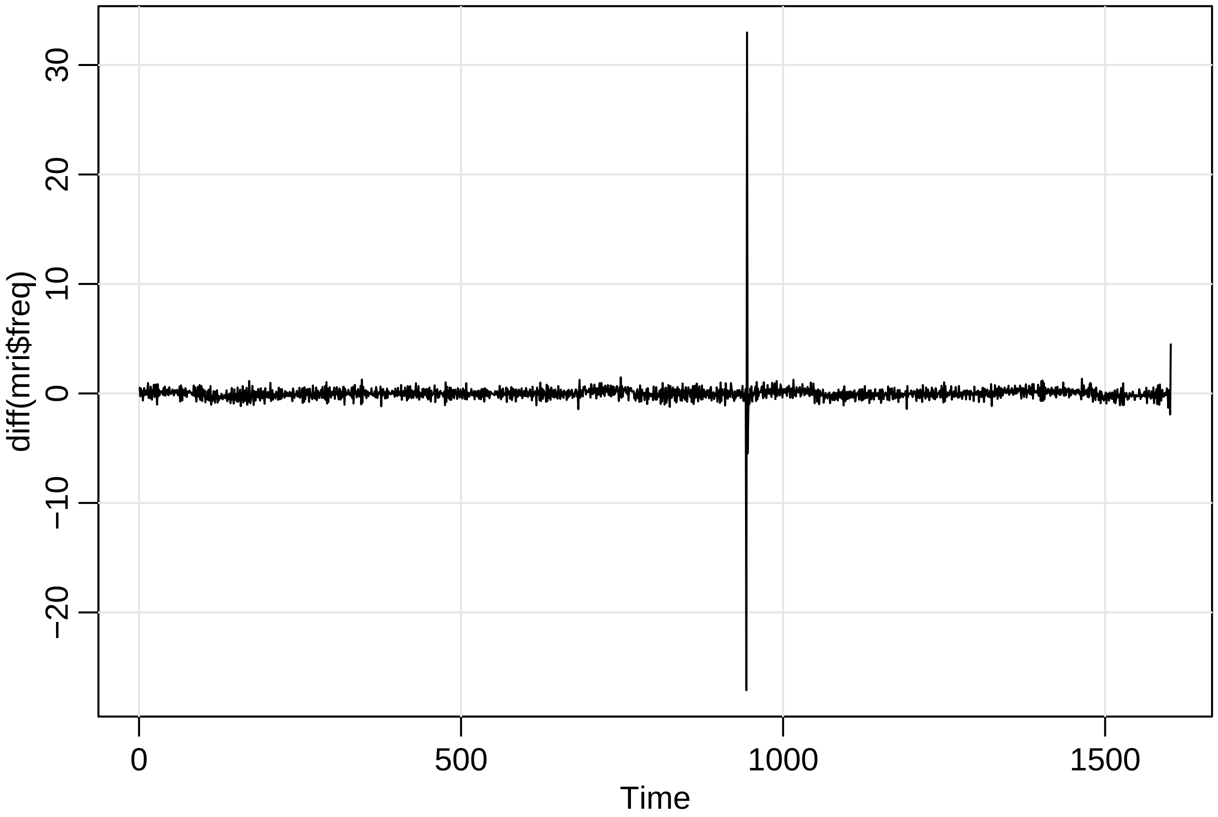


The kernel smoother hugs the original trend line much more closely than the lowess regression. I am going to add the f=0.1 and f=0.2 estimations to the data, because they seem more fair than 0.5 or 1.

```
mri$freq_low_1 <- lowess(mri$freq, f=0.1)$y
mri$freq_low_2 <- lowess(mri$freq, f=0.2)$y
```

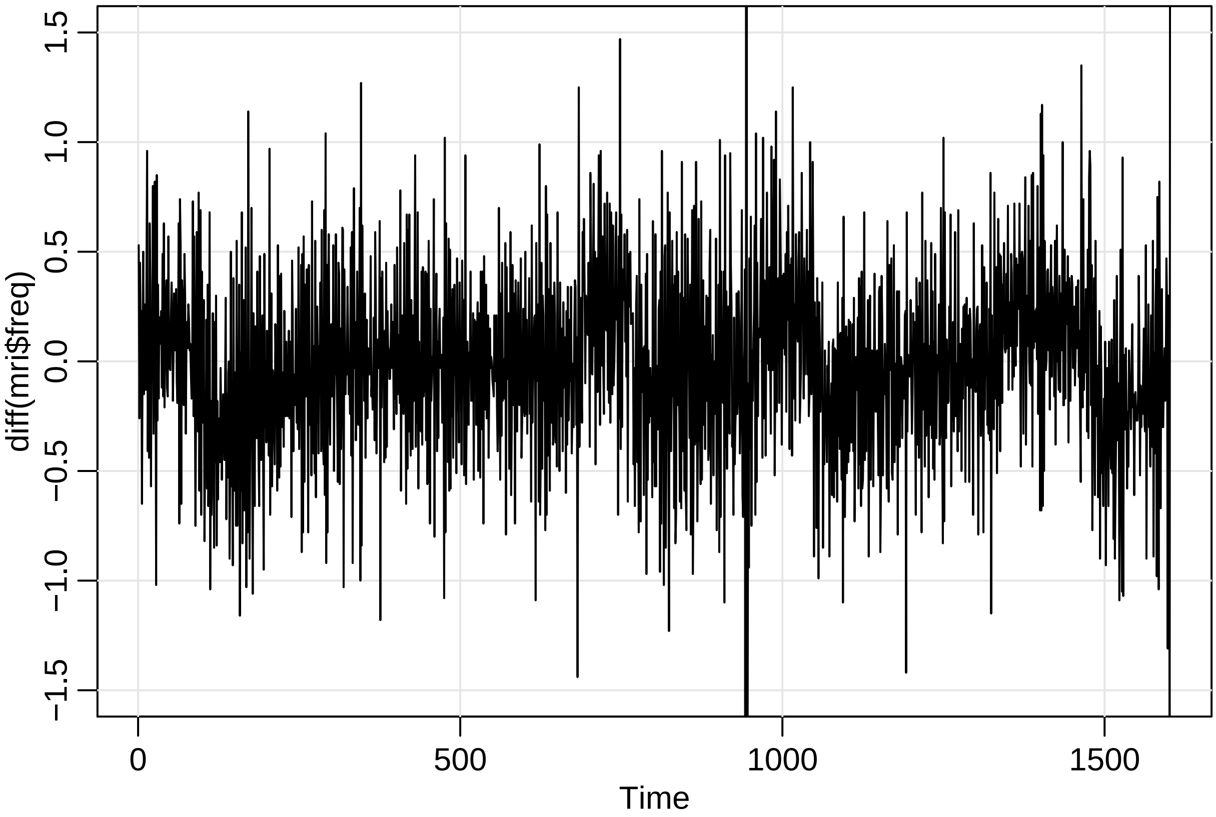
We now have a handful of smoothed data points. Let's see if we can get to stationarity using those as our base. Taking first differences seemed promising in the past so let's do that again.

```
astsa::tsplot(diff(mri$freq))
```



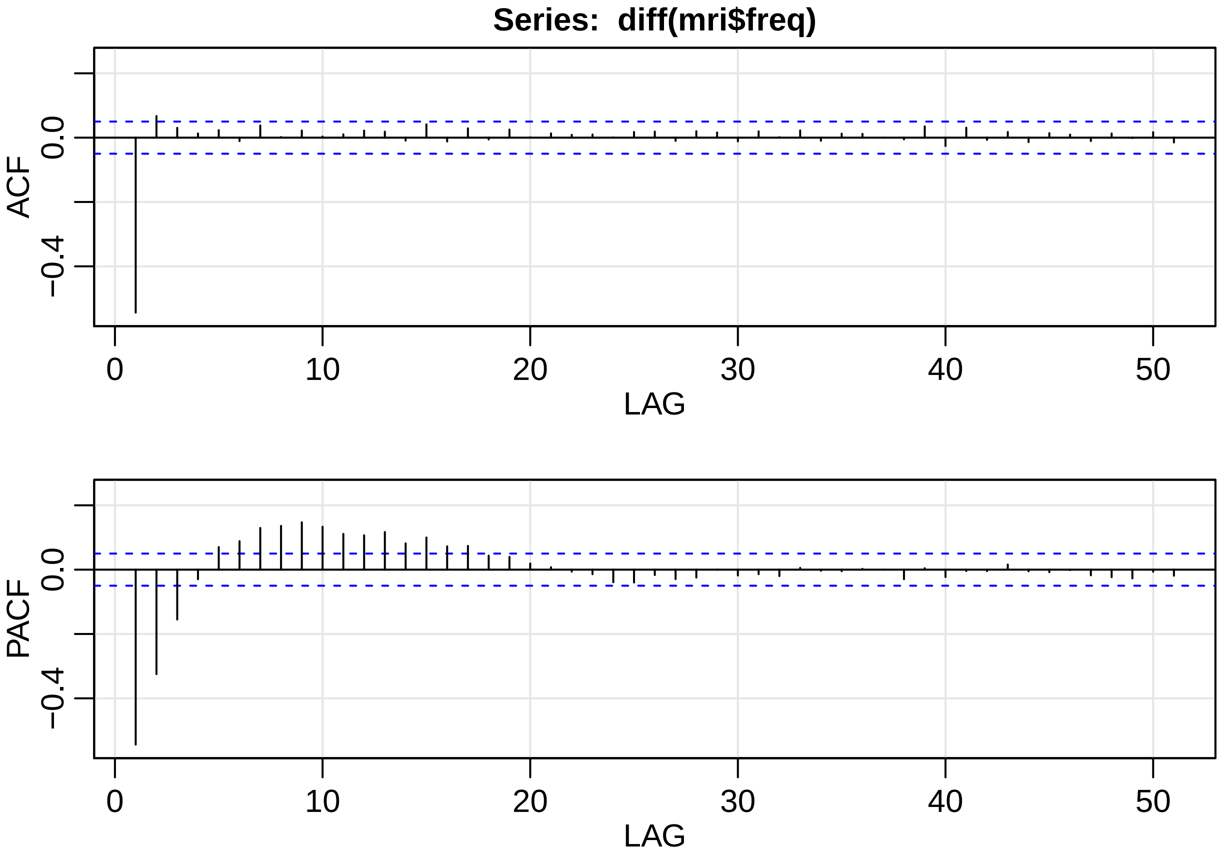
This looks stationary aside from the same problem area, but it may just be an axis thing. Let's adjust it and revisit.

```
astsa::tsplot(diff(mri$freq), ylim=c(-1.5,1.5))
```



There looks to be some autocorrelation when we get up close.

```
acf2(diff(mri$freq))
```

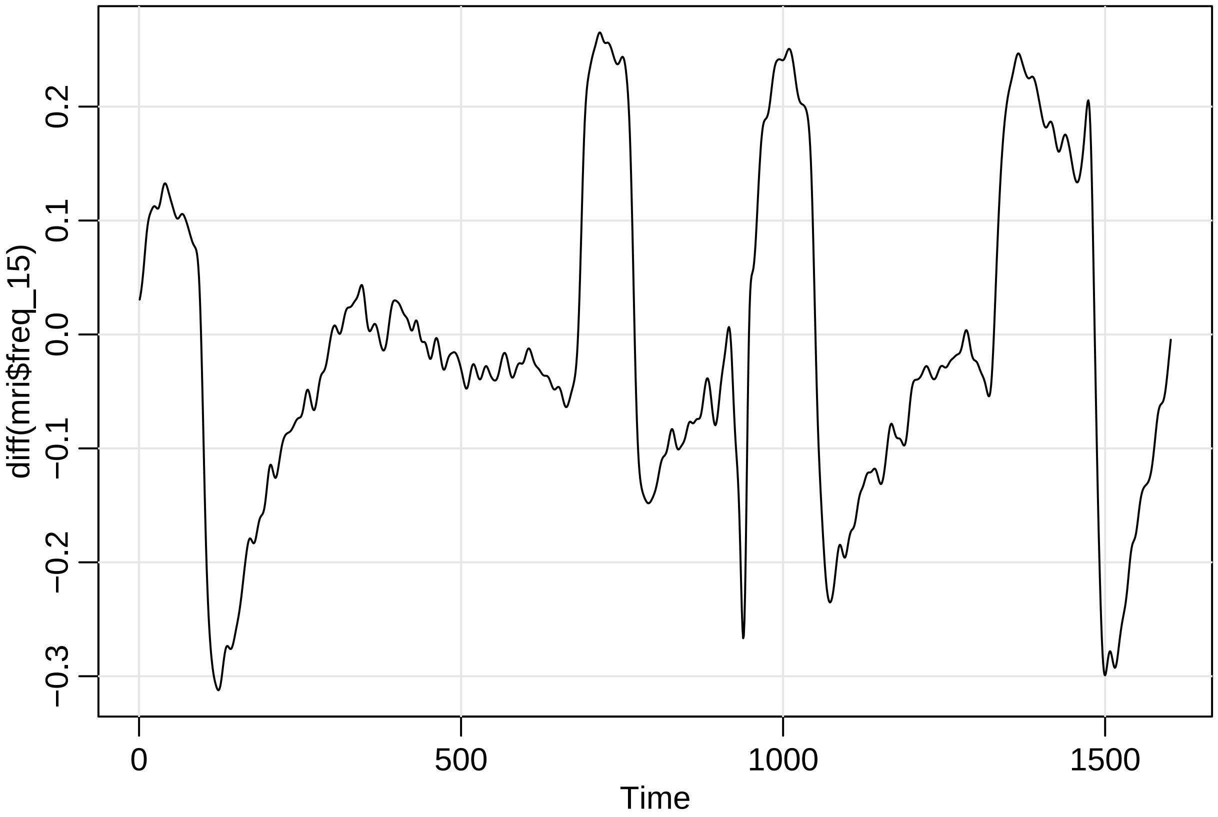


```
##         ACF  PACF
##  [1,] -0.54 -0.54
##  [2,]  0.07 -0.32
##  [3,]  0.03 -0.16
##  [4,]  0.01 -0.03
##  [5,]  0.02  0.07
##  [6,] -0.01  0.09
##  [7,]  0.04  0.13
##  [8,]  0.00  0.14
##  [9,]  0.02  0.15
## [10,]  0.00  0.13
## [11,]  0.01  0.11
## [12,]  0.02  0.11
## [13,]  0.02  0.12
## [14,] -0.01  0.08
## [15,]  0.04  0.10
## [16,] -0.01  0.07
## [17,]  0.03  0.07
## [18,] -0.01  0.04
## [19,]  0.03  0.04
## [20,]  0.00  0.02
## [21,]  0.01  0.01
## [22,]  0.01 -0.01
## [23,]  0.01 -0.01
## [24,]  0.00 -0.04
## [25,]  0.02 -0.04
## [26,]  0.02 -0.02
## [27,] -0.01 -0.03
## [28,]  0.02 -0.02
## [29,]  0.02  0.00
## [30,] -0.01 -0.02
## [31,]  0.02 -0.01
## [32,]  0.00 -0.02
## [33,]  0.02  0.01
## [34,] -0.01  0.00
## [35,]  0.01  0.00
## [36,]  0.01  0.00
## [37,]  0.00  0.00
## [38,] -0.01 -0.03
## [39,]  0.04  0.00
## [40,] -0.03 -0.02
## [41,]  0.03  0.00
## [42,] -0.01  0.00
## [43,]  0.02  0.02
## [44,] -0.01 -0.01
## [45,]  0.01 -0.01
## [46,]  0.01  0.00
## [47,] -0.01 -0.02
## [48,]  0.01 -0.02
## [49,]  0.00 -0.03
## [50,]  0.02 -0.01
## [51,] -0.01 -0.02
```

It's tough to see in the differenced time series plot, but we do have a strong and significant negative lag at lag 1. Perhaps the differenced series is an MA(1) model? We can look at this idea later.

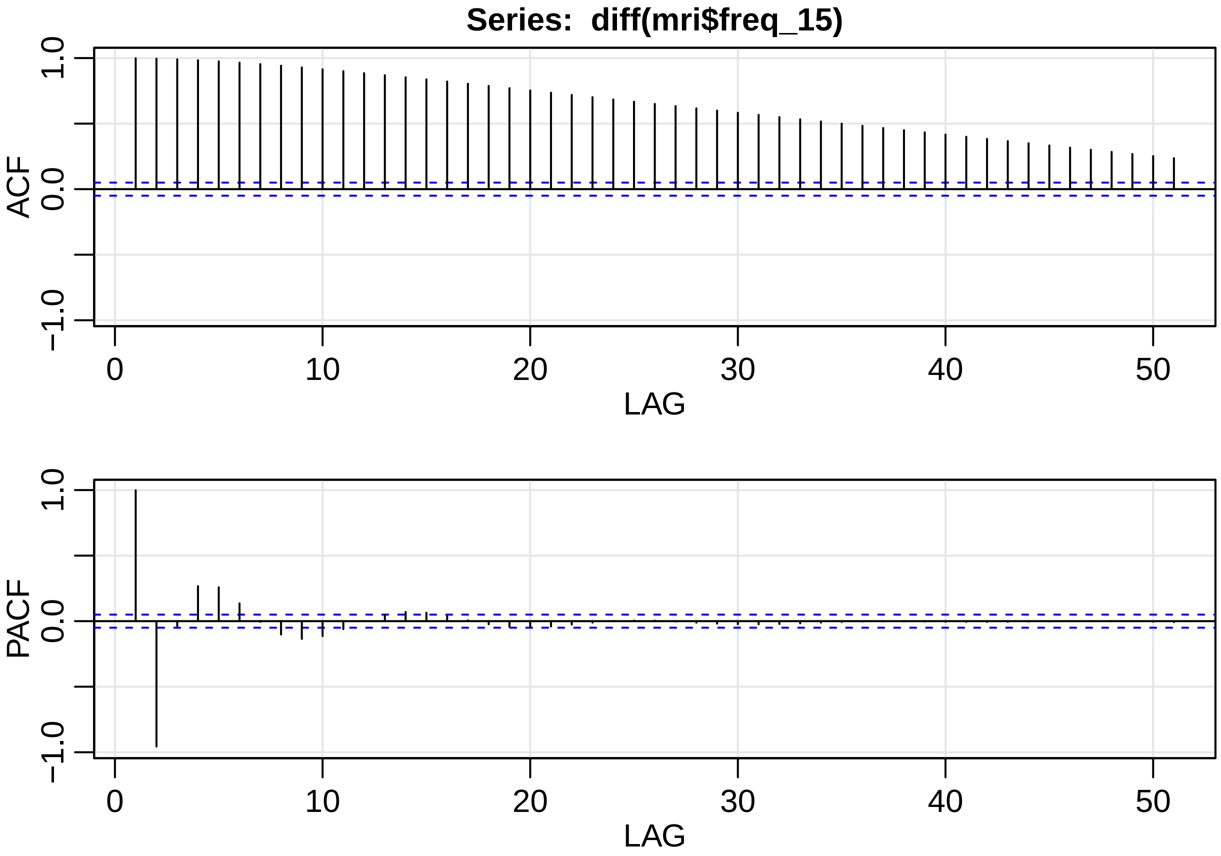
Let's repeat this exercise for some of our smoothed series and see.

```
# kernel smoother with bandwidth of 15
astsa::tsplot(diff(mri$freq_15))
```



Clear trend.

```
acf2(diff(mri$freq_15))
```

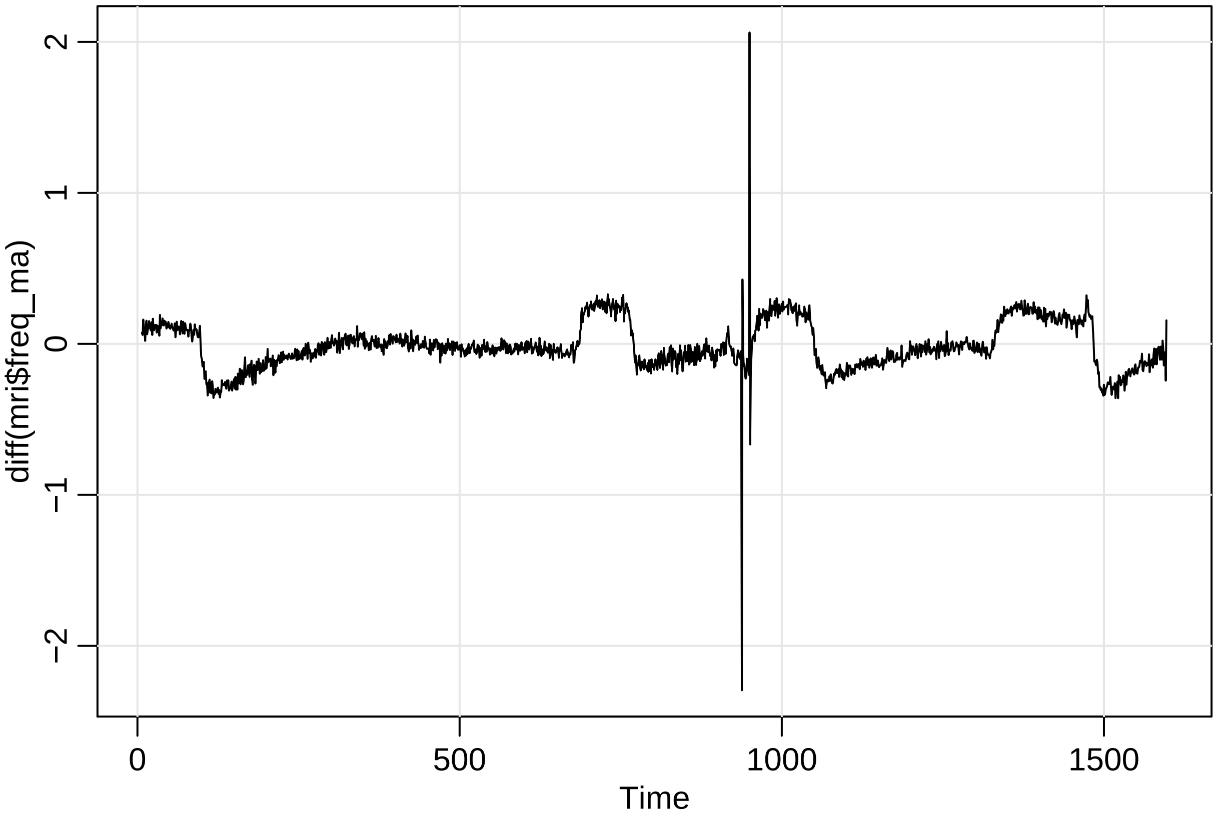


```
##        ACF  PACF
##  [1,] 1.00  1.00
##  [2,] 1.00 -0.96
##  [3,] 0.99 -0.04
##  [4,] 0.98  0.27
##  [5,] 0.98  0.26
##  [6,] 0.97  0.14
##  [7,] 0.95 -0.01
##  [8,] 0.94 -0.10
##  [9,] 0.93 -0.14
## [10,] 0.91 -0.11
## [11,] 0.90 -0.06
## [12,] 0.89  0.00
## [13,] 0.87  0.05
## [14,] 0.85  0.07
## [15,] 0.84  0.06
## [16,] 0.82  0.04
## [17,] 0.80  0.01
## [18,] 0.79 -0.02
## [19,] 0.77 -0.04
## [20,] 0.75 -0.05
## [21,] 0.74 -0.04
## [22,] 0.72 -0.03
## [23,] 0.70 -0.01
## [24,] 0.68  0.00
## [25,] 0.67  0.00
## [26,] 0.65  0.00
## [27,] 0.63  0.00
## [28,] 0.62 -0.01
## [29,] 0.60 -0.02
## [30,] 0.58 -0.02
## [31,] 0.57 -0.02
## [32,] 0.55 -0.02
## [33,] 0.53 -0.02
## [34,] 0.52 -0.01
## [35,] 0.50 -0.01
## [36,] 0.48  0.00
## [37,] 0.47  0.00
## [38,] 0.45  0.00
## [39,] 0.43  0.00
## [40,] 0.42  0.00
## [41,] 0.40 -0.01
## [42,] 0.38 -0.01
## [43,] 0.37 -0.01
## [44,] 0.35  0.00
## [45,] 0.33  0.00
## [46,] 0.32  0.00
## [47,] 0.30  0.00
## [48,] 0.28  0.00
## [49,] 0.27  0.00
## [50,] 0.25  0.00
## [51,] 0.24 -0.01
```

Now suggestive of an AR(p) model.

Let's just try the simpler moving average smoother instead of fancy things.

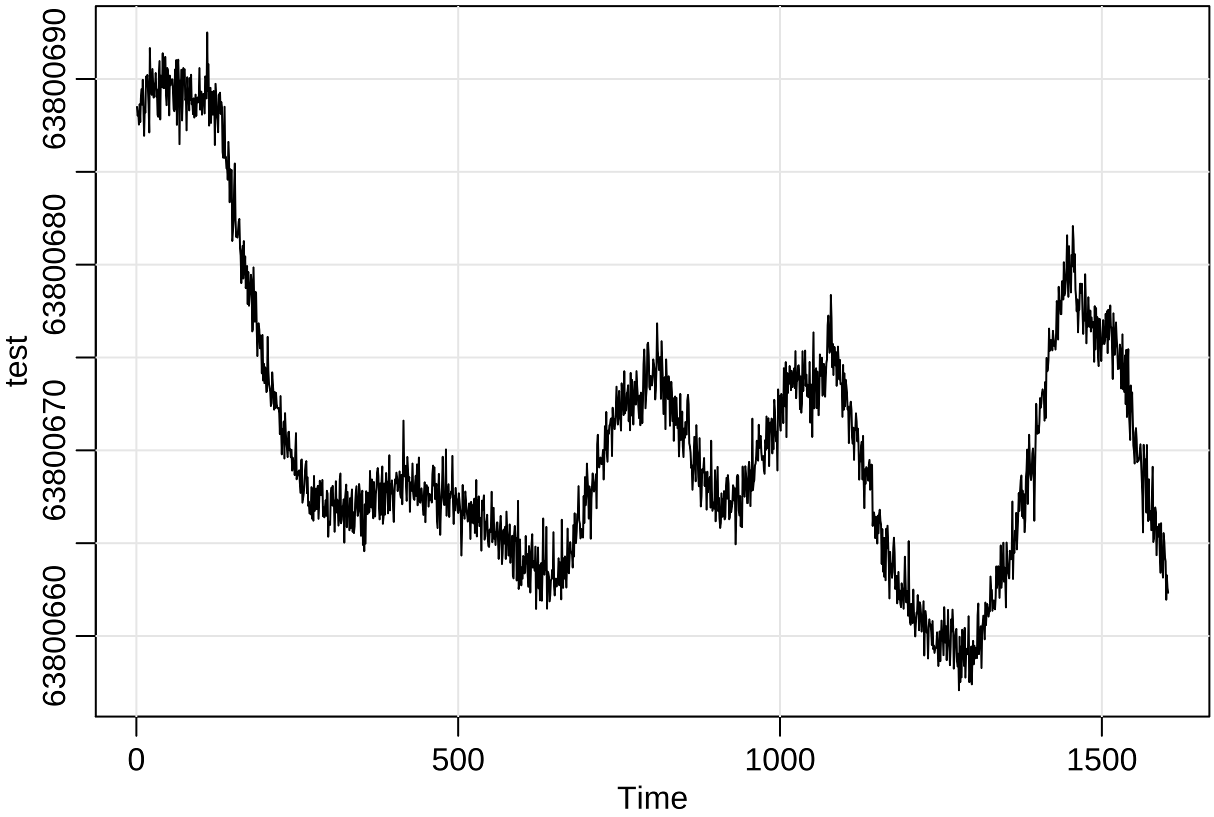
```r
# 12 point MA
mri$freq_ma <- stats::filter(mri$freq, sides=2, filter=rep(1/12, 12))
astsa::tsplot(diff(mri$freq_ma))
```



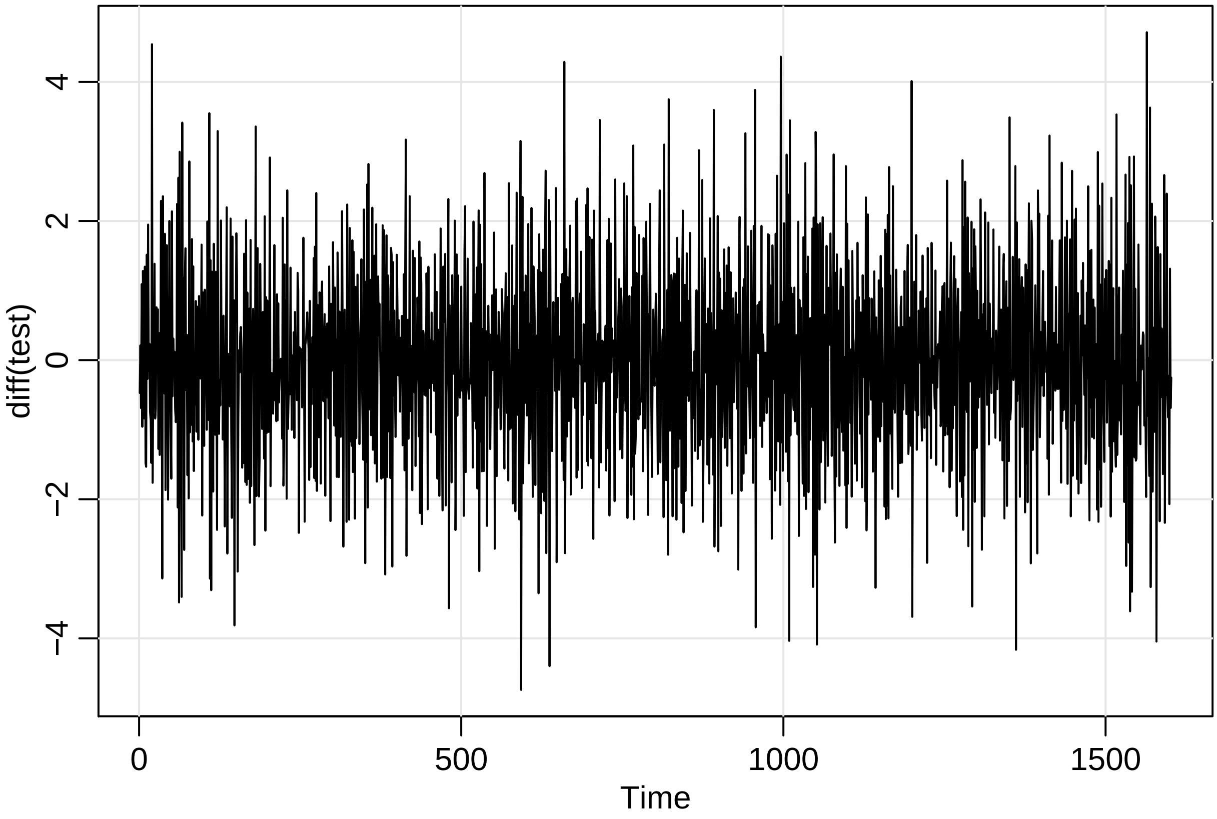
Similar story.

Ironic, but let's add noise to the smoothed series and see if that helps.

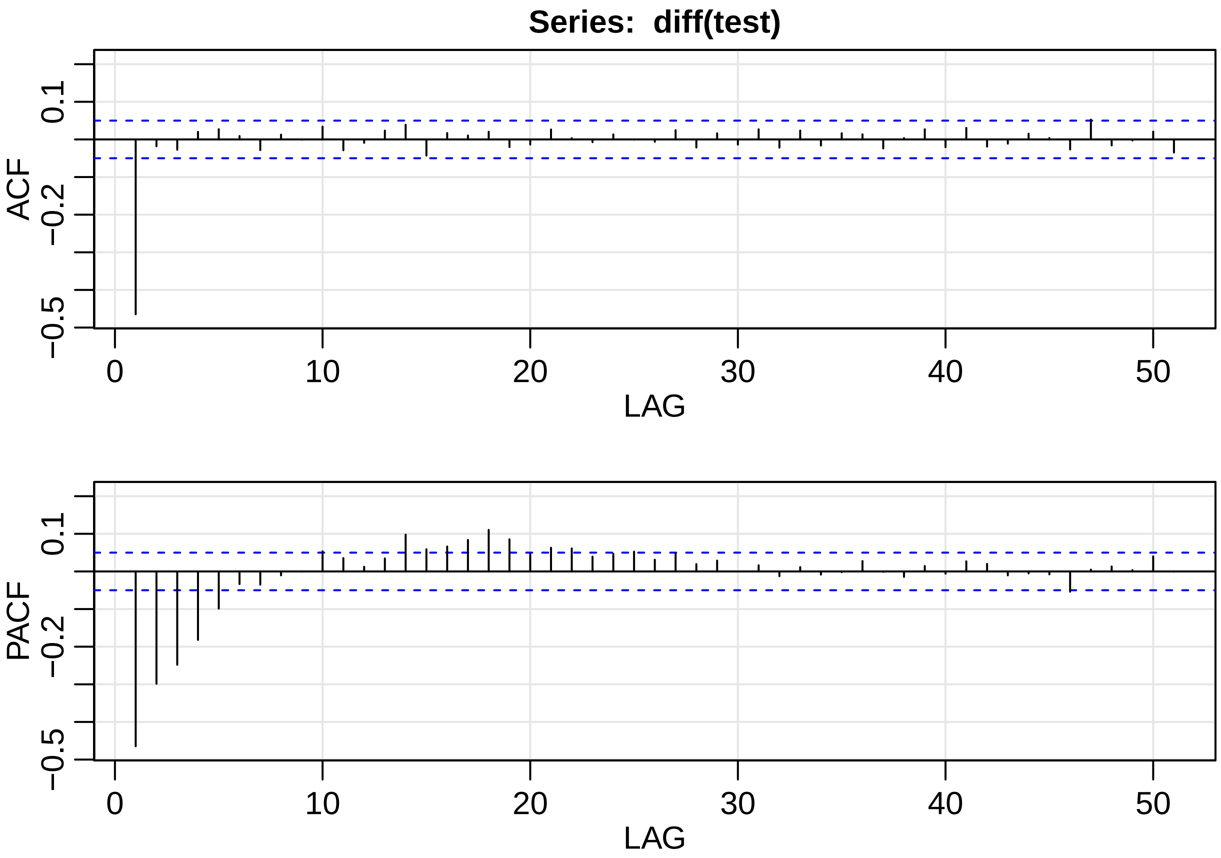
```
test <- mri$freq_low_1 + rnorm(length(mri$freq_point5))
astsa::tsplot(test)
```



```
astsa::tsplot(diff(test))
```



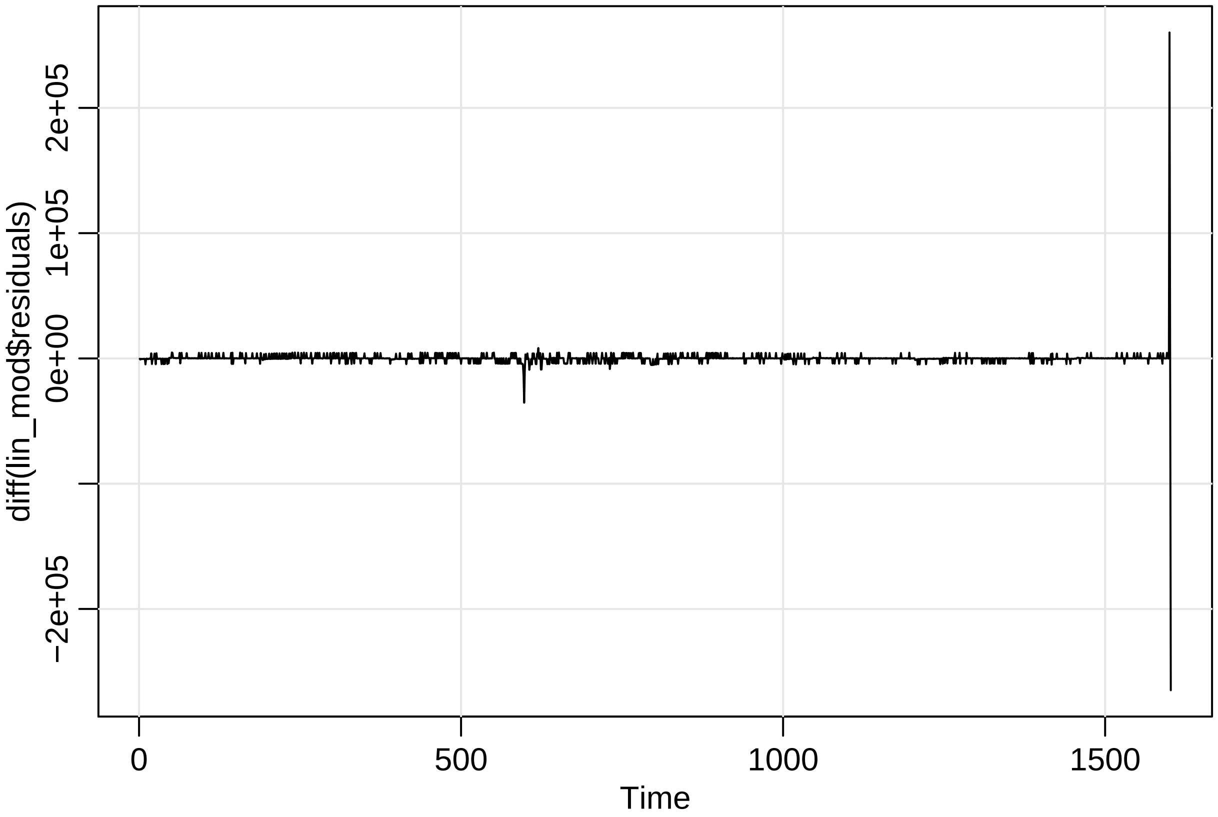
```
acf2(diff(test))
```



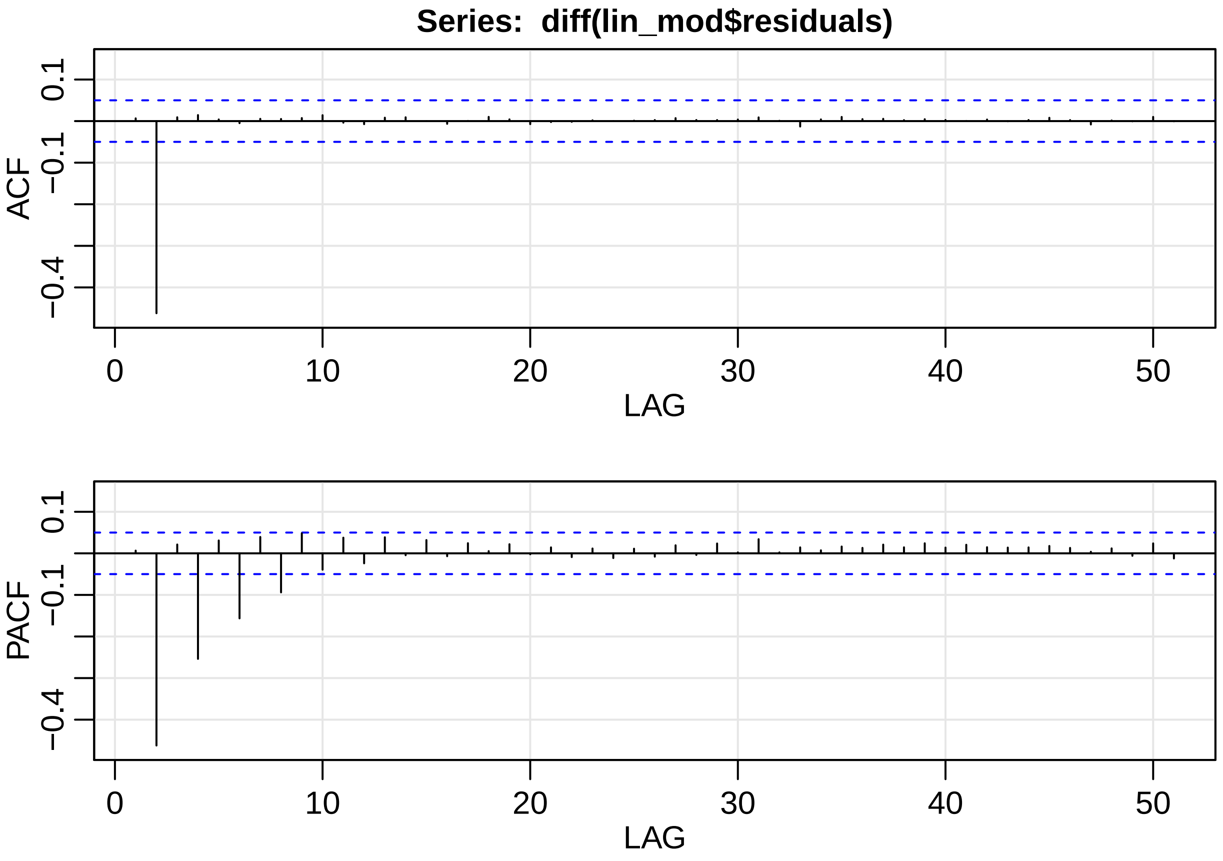
```
##        ACF  PACF
##  [1,] -0.46 -0.46
##  [2,] -0.02 -0.30
##  [3,] -0.03 -0.25
##  [4,]  0.02 -0.18
##  [5,]  0.03 -0.10
##  [6,]  0.01 -0.03
##  [7,] -0.03 -0.04
##  [8,]  0.01 -0.01
##  [9,]  0.00  0.00
## [10,]  0.03  0.05
## [11,] -0.03  0.04
## [12,] -0.01  0.01
## [13,]  0.02  0.03
## [14,]  0.04  0.10
## [15,] -0.04  0.06
## [16,]  0.02  0.07
## [17,]  0.01  0.08
## [18,]  0.02  0.11
## [19,] -0.02  0.09
## [20,] -0.01  0.05
## [21,]  0.03  0.06
## [22,]  0.00  0.06
## [23,] -0.01  0.04
## [24,]  0.01  0.05
## [25,]  0.00  0.05
## [26,] -0.01  0.03
## [27,]  0.02  0.05
## [28,] -0.02  0.02
## [29,]  0.02  0.03
## [30,] -0.01  0.00
## [31,]  0.03  0.02
## [32,] -0.02 -0.01
## [33,]  0.02  0.01
## [34,] -0.02 -0.01
## [35,]  0.02  0.00
## [36,]  0.01  0.03
## [37,] -0.02  0.00
## [38,]  0.00 -0.01
## [39,]  0.03  0.01
## [40,] -0.02 -0.01
## [41,]  0.03  0.03
## [42,] -0.02  0.02
## [43,] -0.01 -0.01
## [44,]  0.02 -0.01
## [45,]  0.00 -0.01
## [46,] -0.03 -0.05
## [47,]  0.05  0.00
## [48,] -0.02  0.01
## [49,]  0.00  0.00
## [50,]  0.02  0.04
## [51,] -0.03  0.00
```

That was promising for a brief second. Let's try *detrending* instead of differencing.

```
lin_mod <- lm(mri$freq ~ 0 + mri$atm_pressure + mri$int_pressure)
astsa::tsplot(diff(lin_mod$residuals))
```



```
acf2(diff(lin_mod$residuals))
```

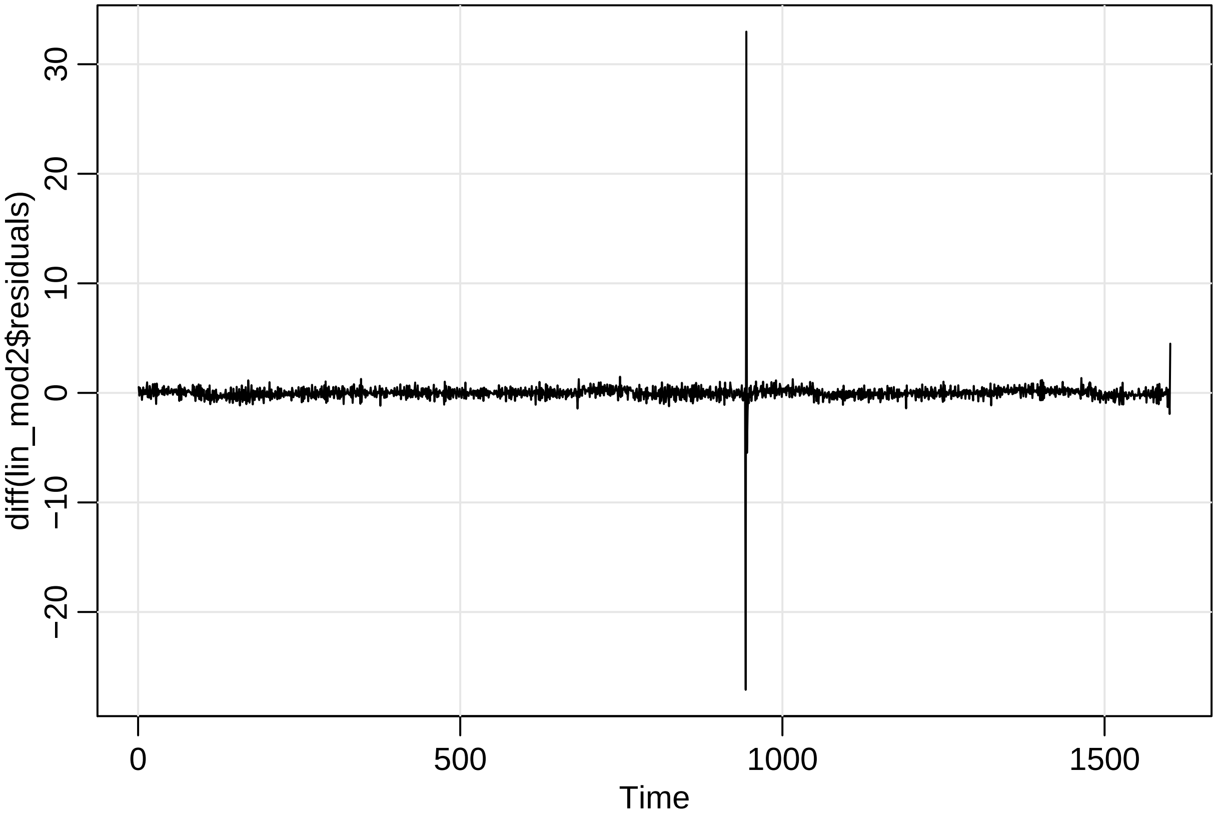


```
##         ACF  PACF
##  [1,]  0.01  0.01
##  [2,] -0.46 -0.46
##  [3,]  0.01  0.02
##  [4,]  0.01 -0.25
##  [5,]  0.00  0.03
##  [6,]  0.00 -0.16
##  [7,]  0.01  0.04
##  [8,]  0.01 -0.09
##  [9,]  0.01  0.05
## [10,]  0.01 -0.04
## [11,]  0.00  0.04
## [12,] -0.01 -0.02
## [13,]  0.01  0.04
## [14,]  0.01  0.00
## [15,]  0.00  0.03
## [16,] -0.01 -0.01
## [17,]  0.00  0.02
## [18,]  0.01  0.01
## [19,]  0.00  0.02
## [20,] -0.01  0.00
## [21,]  0.00  0.01
## [22,]  0.00 -0.01
## [23,]  0.00  0.01
## [24,]  0.00 -0.01
## [25,]  0.00  0.01
## [26,]  0.00 -0.01
## [27,]  0.01  0.02
## [28,]  0.00  0.00
## [29,]  0.00  0.02
## [30,]  0.00  0.00
## [31,]  0.01  0.03
## [32,]  0.00  0.00
## [33,] -0.01  0.01
## [34,]  0.00  0.01
## [35,]  0.01  0.02
## [36,]  0.00  0.01
## [37,]  0.01  0.02
## [38,]  0.00  0.01
## [39,]  0.00  0.02
## [40,]  0.00  0.01
## [41,]  0.00  0.02
## [42,]  0.00  0.01
## [43,]  0.00  0.01
## [44,]  0.00  0.01
## [45,]  0.01  0.02
## [46,]  0.00  0.01
## [47,] -0.01  0.00
## [48,]  0.00  0.01
## [49,]  0.00 -0.01
## [50,]  0.01  0.02
## [51,]  0.00 -0.01
```

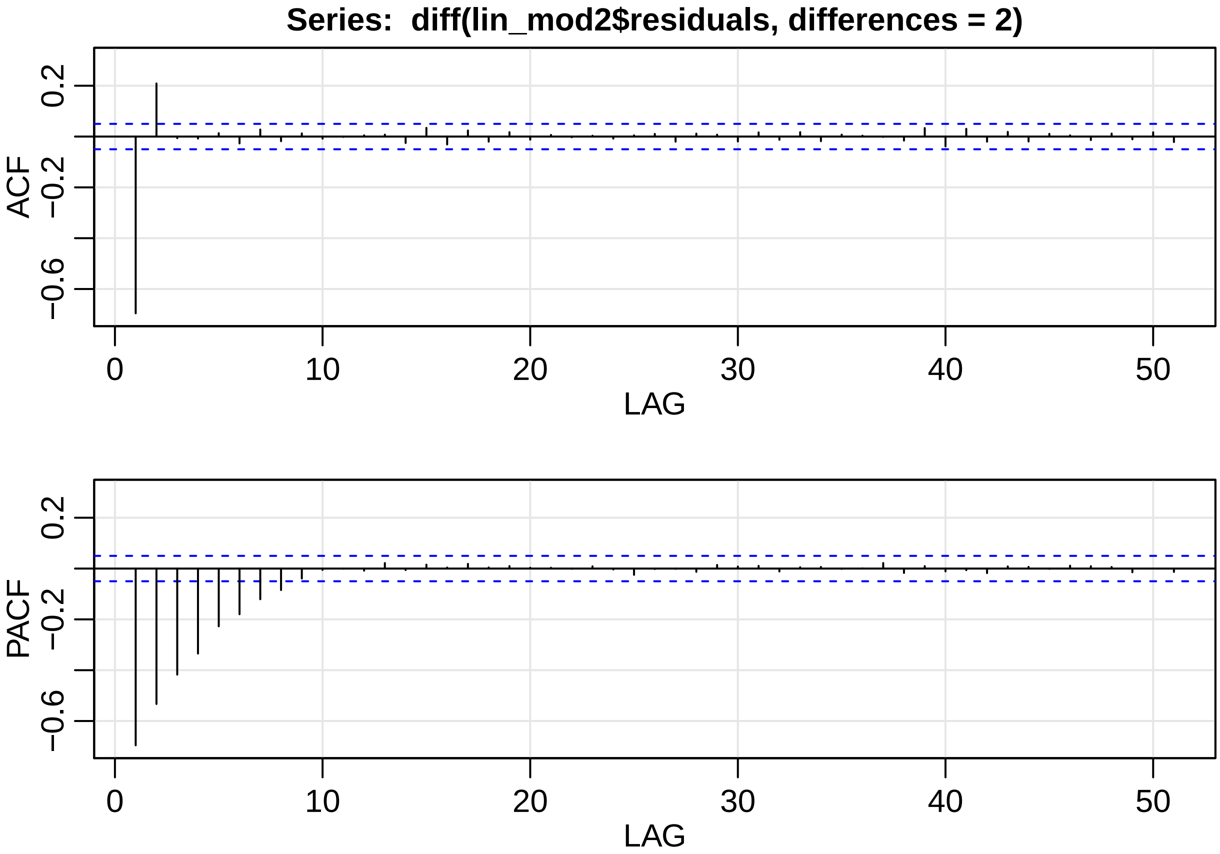
Note that the MA(q) model ACF cuts off after lag q, while the PACF tails off. This could related to that.

Detrend with one of the smoothers.

```
lin_mod2 <- lm(mri$freq_point5 ~ time(mri$freq_point5))
astsa::tsplot(diff(lin_mod2$residuals))
```



```
acf2(diff(lin_mod2$residuals, differences=2))
```

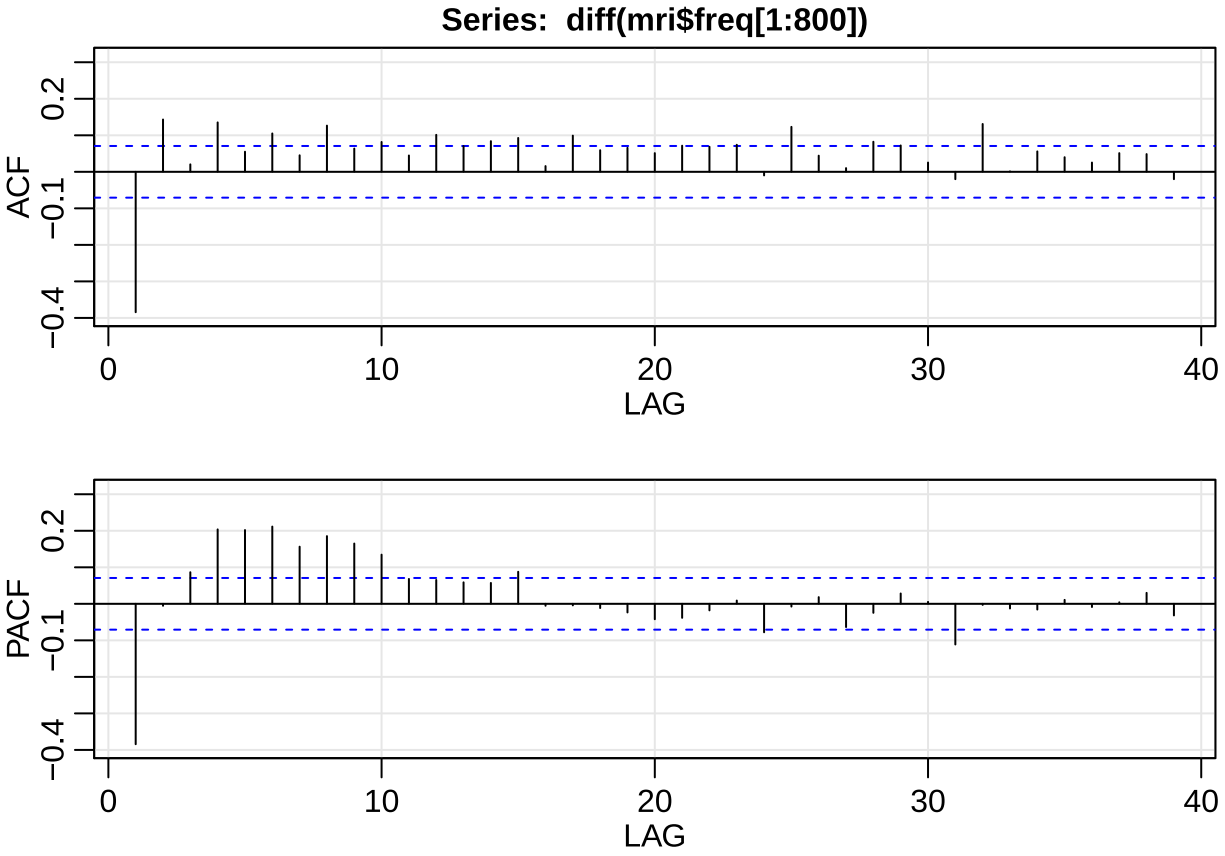


```
##          ACF  PACF
##  [1,] -0.70 -0.70
##  [2,]  0.21 -0.53
##  [3,] -0.01 -0.42
##  [4,] -0.01 -0.33
##  [5,]  0.01 -0.23
##  [6,] -0.03 -0.18
##  [7,]  0.03 -0.12
##  [8,] -0.02 -0.08
##  [9,]  0.01 -0.04
## [10,] -0.01 -0.01
## [11,]  0.00  0.00
## [12,]  0.00 -0.01
## [13,]  0.01  0.02
## [14,] -0.03 -0.01
## [15,]  0.03  0.02
## [16,] -0.03  0.00
## [17,]  0.02  0.02
## [18,] -0.02  0.00
## [19,]  0.02  0.01
## [20,] -0.01  0.00
## [21,]  0.01  0.00
## [22,]  0.00  0.00
## [23,]  0.00  0.01
## [24,] -0.01  0.00
## [25,]  0.00 -0.02
## [26,]  0.01  0.00
## [27,] -0.02  0.00
## [28,]  0.01 -0.01
## [29,]  0.01  0.01
## [30,] -0.02  0.01
## [31,]  0.02  0.01
## [32,] -0.01 -0.01
## [33,]  0.02  0.01
## [34,] -0.02  0.01
## [35,]  0.01  0.00
## [36,]  0.00  0.00
## [37,]  0.00  0.02
## [38,] -0.02 -0.02
## [39,]  0.03  0.01
## [40,] -0.04 -0.01
## [41,]  0.03 -0.01
## [42,] -0.02 -0.02
## [43,]  0.02  0.01
## [44,] -0.02  0.01
## [45,]  0.01  0.00
## [46,]  0.00  0.01
## [47,] -0.01  0.01
## [48,]  0.01  0.01
## [49,] -0.01 -0.01
## [50,]  0.02  0.00
## [51,] -0.02 -0.01
```

Suggests MA(2) model.

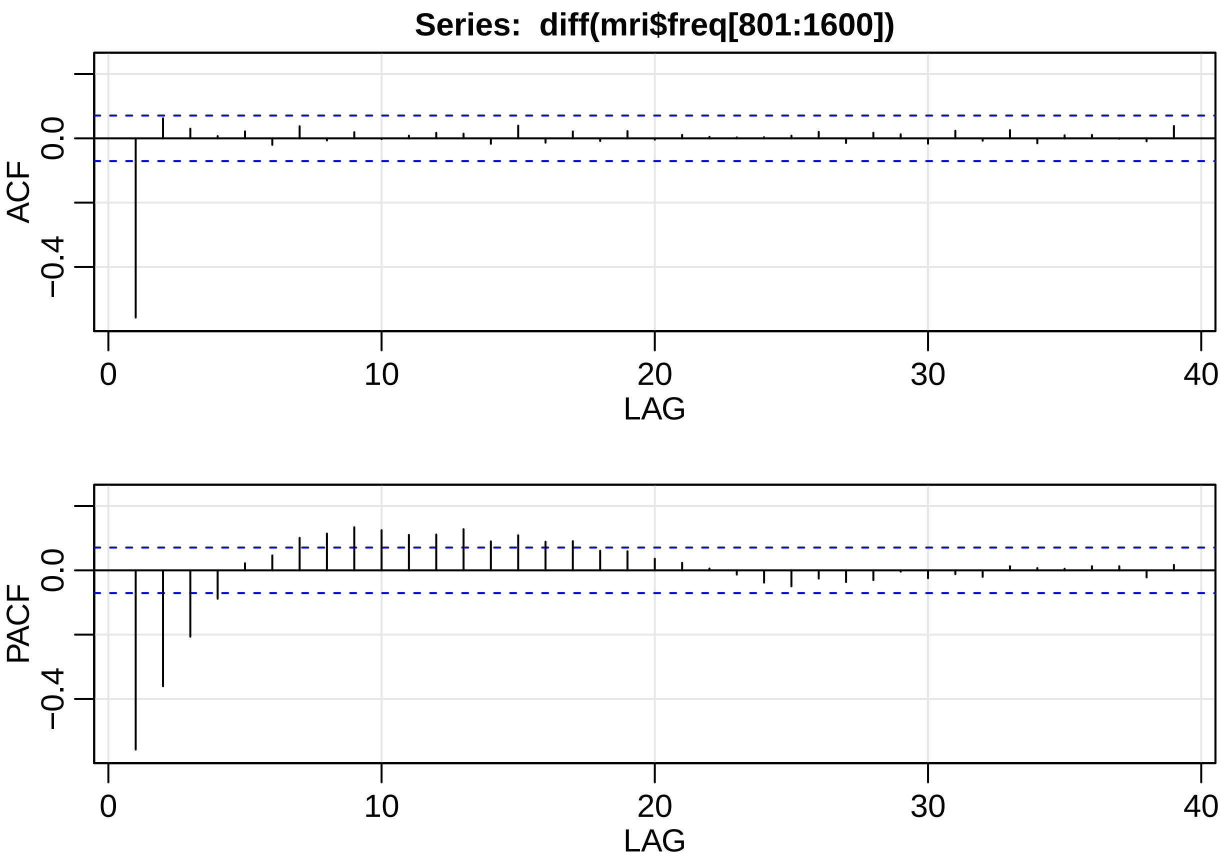
Split the series into halves and see if anything changes.

```
acf2(diff(mri$freq[1:800]))
```



```
##          ACF  PACF
##  [1,] -0.38 -0.38
##  [2,]  0.14 -0.01
##  [3,]  0.02  0.09
##  [4,]  0.14  0.20
##  [5,]  0.05  0.20
##  [6,]  0.11  0.21
##  [7,]  0.05  0.16
##  [8,]  0.13  0.19
##  [9,]  0.06  0.17
## [10,]  0.08  0.13
## [11,]  0.04  0.07
## [12,]  0.10  0.07
## [13,]  0.07  0.06
## [14,]  0.08  0.06
## [15,]  0.09  0.09
## [16,]  0.02 -0.01
## [17,]  0.10  0.00
## [18,]  0.06 -0.01
## [19,]  0.07 -0.02
## [20,]  0.05 -0.04
## [21,]  0.07 -0.04
## [22,]  0.07 -0.02
## [23,]  0.07  0.01
## [24,] -0.01 -0.08
## [25,]  0.12 -0.01
## [26,]  0.04  0.02
## [27,]  0.01 -0.06
## [28,]  0.08 -0.02
## [29,]  0.07  0.03
## [30,]  0.03  0.01
## [31,] -0.02 -0.11
## [32,]  0.13  0.00
## [33,]  0.00 -0.01
## [34,]  0.06 -0.02
## [35,]  0.04  0.01
## [36,]  0.03 -0.01
## [37,]  0.05  0.00
## [38,]  0.05  0.03
## [39,] -0.02 -0.03
```

```
acf2(diff(mri$freq[801:1600]))
```



```
##          ACF  PACF
##  [1,] -0.56 -0.56
##  [2,]  0.06 -0.36
##  [3,]  0.03 -0.21
##  [4,]  0.01 -0.09
##  [5,]  0.02  0.02
##  [6,] -0.02  0.05
##  [7,]  0.04  0.10
##  [8,] -0.01  0.11
##  [9,]  0.02  0.13
## [10,]  0.00  0.12
## [11,]  0.01  0.11
## [12,]  0.02  0.11
## [13,]  0.01  0.13
## [14,] -0.02  0.09
## [15,]  0.04  0.11
## [16,] -0.01  0.09
## [17,]  0.02  0.09
## [18,] -0.01  0.06
## [19,]  0.02  0.06
## [20,]  0.00  0.04
## [21,]  0.01  0.02
## [22,]  0.00  0.01
## [23,]  0.00 -0.01
## [24,]  0.00 -0.04
## [25,]  0.01 -0.05
## [26,]  0.02 -0.03
## [27,] -0.01 -0.04
## [28,]  0.02 -0.03
## [29,]  0.01  0.00
## [30,] -0.02 -0.02
## [31,]  0.02 -0.01
## [32,] -0.01 -0.02
## [33,]  0.03  0.01
## [34,] -0.02  0.01
## [35,]  0.01  0.00
## [36,]  0.01  0.01
## [37,]  0.00  0.01
## [38,] -0.01 -0.02
## [39,]  0.04  0.02
```

Try structural modeling.

```r
decompose(mri$freq_low_2)
```

```
## Error in decompose(mri$freq_low_2): time series has no or less than 2 periods
```

Good to know.